

將 nice 複製到自己的目錄下,名為 nice_pro,必且讓 nice_pro 擁有提高優先權的能力

```
myp@SP:~/system-programming/ch08$ whereis nice
nice: /usr/bin/nice /usr/share/man/man1/nice.1.gz /usr/share/man/man2/nice.2.gz
myp@SP:~/system-programming/ch08$ cp /usr/bin/nice nice-pro
myp@SP:~/system-programming/ch08$ ./nice-pro -n -10 ls
bash: ./nice-pro: No such file or directory
myp@SP:~/system-programming/ch08$ ./nice-pro -n -10 ls
./nice-pro: cannot set niceness: Permission denied
'2021 作業08.pdf'    atexit.c    echo        hw8         myNice      on_exit.c
408410056.tar.bz2   chown_super echo.c       listEnv     myNice.c    vfork8
a.out               cpu_set    getEnv      listEnv.c   nice-pro    vfork8.c
atexit             cpu_set.c  getEnv.c    makefile    on_exit
myp@SP:~/system-programming/ch08$ sudo setcap CAP_SYS_NICE+ep ./nice-pro
[sudo] password for myp:
myp@SP:~/system-programming/ch08$ ./nice-pro -n -10 ls
'2021 作業08.pdf'    atexit.c    echo        hw8         myNice      on_exit.c
408410056.tar.bz2   chown_super echo.c       listEnv     myNice.c    vfork8
a.out               cpu_set    getEnv      listEnv.c   nice-pro    vfork8.c
atexit             cpu_set.c  getEnv.c    makefile    on_exit
myp@SP:~/system-programming/ch08$
```

1. 先用whereis nice找出nice的位置
2. 將/usr/bin/nice複製到自己的目錄下(nice-pro)
3. 先執行一次./nice-pro -n -10 ls('-10' 提高nice-pro優先權)會發現permission denied
4. 之後輸入sudo setcap CAP_SYS_NICE+ep ./nice-pro讓 nice-pro擁有部分權限
5. 再執行一次./nice-pro -n -10 ls就不會出現permission denied的情況

從 man capabilities 裡面隨便挑三個權限,並說明那三個權限是什麼樣的用途

CAP_CHOWN 修改文件所有者的權限

CAP_MAC_ADMIN 允許 MAC 配置或狀態更改

CAP_SYS_BOOT 允許重新啟動系統

想辦法量測『優先權高一等級的 task 比正常優先權的 task 速度快多少』?

撰寫一個alarm.c程式，使用fork()產生出父子行程，並由父子行程印出cpp值的大小判斷速度

```
#include <stdio.h>
#include <unistd.h>
#include <errno.h>
#include <stdlib.h>
#include <signal.h>
long long int cpp = 0;
void alarmHandler(int signo)
{
    printf("cpp = %lld\n", cpp);
    exit(0);
}
int main(int argc, char **argv)
{
    int nice_v = atoi(argv[1]); //讀入 nice 的參數
    int childPid = fork();      //產生二個行程
    if (childPid > 0)
    {
        //養我們的雙親
        nice(nice_v); //修改 parent 的 nice 值
    }
    else
    {
        //child 不用特別處理
    }
    //底下的程式碼無論 child 是否大於 0 都會執行
    //設定鬧鐘(SIGALRM)叫的時候，『作業系統』呼叫alarmHandler
    signal(SIGALRM, alarmHandler);
    //把鬧鐘的時間定在 1 秒鐘以後
    alarm(1);
    //不斷地 cpp++
    while (1)
    {
        cpp++;
    }
}
```

如果設定比正常優先權高一等級的優先權，執行結果如下：

```
mosp@SP:~/system-programming/ch08$ ./a.out 1  
cpp = 117905264  
mosp@SP:~/system-programming/ch08$ cpp = 147408638
```

$147408638 \div 117905264 =$

1.25022948933

把cpp值丟去計算機得到差一等級速度差異為1.25倍